CLAIM LISTING

1. (Currently Amended) A true random number generator comprising:

(a) a first oscillator producing a first oscillatory signal [producing means];

(b) a second oscillator producing a second oscillatory signal [producing means];

(c) a frequency multiplier [multiplication means] responsive to said second oscillatory signal [producing means];

(d) a processor [means] responsive to said first oscillatory signal [producing means] and said frequency multiplier [multiplication means], to produce a sequence of true random numbers.

2. (Currently Amended) A true random number generator as described in Claim 1 wherein said frequency multiplier [multiplication means] is selected from the group consisting of: a phase locked loop frequency multiplier, a nonlinear frequency multiplier, and an exclusive-or logic function with unequal time delay elements on its inputs.

3. (Currently amended) A true random number generator as described in Claim 1 wherein the average output frequency of said frequency multiplier [multiplication means] is at least 10,000 times the average frequency of said first oscillatory signal.

4. (Currently amended) A true random number generator as described in Claim 1 wherein [the] said processor [means] includes a counter which is incremented in response to the output of said frequency multiplier [multiplication means], said counter having a count that is read in response to said first oscillatory signal [producing means].

5. (Original) A true random number generator comprising:

(a) a personal computer including an oscillator and a first frequency multiplier responsive to said first oscillator;

(b) a microprocessor included within said personal computer, said microprocessor including a second frequency multiplier, and responsive to said oscillator and said first and second frequency multipliers to produce a sequence of true random numbers.

6. (Original) A true random number generator as described in Claim 5 that can

produce at least 200 bits of entropy per second.

7. (Original) A true random number generator comprising:

(a) a personal computer, including a first oscillator for producing a low-frequency signal and a second oscillator that is independent of said first oscillator;

(b) a first frequency multiplier responsive to said second oscillator for producing a medium-frequency signal;

(c) a programmable interrupt controller included in said personal computer responsive to said low-frequency signal;

(d) a second frequency multiplier responsive to said medium-frequency signal for producing a high-frequency signal;

(e) a counter responsive to said high-frequency signal;

(f) a microprocessor including said counter and said second frequency multiplier, and responsive to said programmable interrupt controller and said counter to produce a sequence of true random numbers.

8. (Original) A true random number generator as described in Claim 7 wherein said first oscillator is included in one of the personal computer components selected from the group consisting of: a real time clock, a display card, an audio card, a network card, a modem, and a serial port card.

9. (Original) A true random number generator as described in Claim 7 wherein the average output frequency of said high-frequency signal is at least 10,000 times the average frequency of said low-frequency signal.

10. (Original) A true random number generator as described in Claim 7 that can produce at least of 200 bits of entropy per second.

11. (Currently amended) A method of generating true random numbers comprising [the steps of]:

generating a first oscillatory signal;

generating a second oscillatory signal;

determining values of transition jitter included in said first and said second oscillatory signals;

using said values of transition jitter to calculate the entropy available from said first and said second oscillatory signals; and processing said first and said second oscillatory signals to extract said entropy in the form of a sequence of true random numbers.

12. (Currently amended) The method as described in Claim 11 including [the additional step of] processing said sequence of true random numbers to reduce defects in the randomness properties of said sequence of true random numbers.

13. (Currently amended) The method as described in Claim 12 wherein said [step of] processing of said sequence of true random numbers to reduce defects is accomplished by:

establishing three or more circular buffers in a memory system, each of said circular buffers having a length that is relatively prime to all the other buffers;

reading the contents stored at an address from each of said three or more circular buffers;

combining said contents into a resultant number using a first exclusive-or function;

providing said resultant number as one number in a defect-reduced output sequence and also applying said resultant number to the first input of a second exclusive-or function;

applying a number from the defect-containing random number sequence to the second input of said second exclusive-or function;

using the resulting number of said second exclusive-or function to replace one of the numbers stored in one of said circular buffers;

incrementing all the addresses to read from each of said three or more circular buffers, and repeating the steps of reading, combining in said first exclusive-or function and providing a defect-reduced output number, thereby producing a defect-reduced sequence of any desired length and;

repeating the steps of performing said second exclusive-or function on each new defect-reduced output number with a new defect-containing true random number and

using the resulting numbers to sequentially replace each of the numbers stored in each of said three or more circular buffers.

14. (Currently amended) The method as described in Claim 12 wherein said processing to reduce defects in the randomness properties of the true random

sequence is accomplished by [the steps of]:

using a pseudorandom number generator to produce a random number;

providing said random number as a defect-reduced output number;

applying said defect-reduced output number to the first input of an exclusive-or function;

applying a number from a defect-containing true random number sequence to the second input of said exclusive-or function;

using the output of said exclusive-or function as a seed to be used in said pseudorandom number generator and;

repeating the step of generating a defect-reduced output number, each time using another number from the defect-containing true random number sequence and the new seed from said exclusive-or function, thereby producing a defect-reduced random output sequence.

15. (Currently amended) The method as described in Claim 11 wherein calculating the entropy available is accomplished by [the steps of]:

combining all independent transition jitter values into an effective value;

calculating a normalized jitter value by dividing said effective value by the period of said second oscillatory signal;

utilizing said normalized jitter value to calculate an average probability, p(1), that the state of said second oscillatory signal will be high, or 1, when it is predicted to be high at a positive transition of said first oscillatory signal and;

utilizing said average probability, p(1), to calculate the entropy or H using the equation, $H = -(1/\mathrm{Ln}[2])\ (p(1)\ \mathrm{Ln}[p(1)] + (1 - p(1))\ \mathrm{Ln}[1 - p(1)])$.